**Meal-Prep System and Unit Tests**

Remote Gang, 3/7/2022

**System Test Scenarios**

* User story 1 from sprint 2: As a new user I want to create an account on the website.
  + 1. Visit Meal-Prep website; click the signup button on the nav bar and type
    - Username = “newuser1”
    - Email = “user@gmail.com”
    - Password = “password1”
    - Confirm Password = “password1”
    - User should see a verification message telling them to check email to activate their account
  + 2. Go to your email and click the link in the email.
  + 3. Link directs the user to a page with verification message that the account is now active and redirects them to the homepage.
* User story 2 from sprint 2: As a user with an account, I would like to sign into my account.
  + 1. Visit Meal-Prep website; click the login button on the nav bar and type
    - Username = “newuser1” or Email = “[user@gmail.com](mailto:user@gmail.com)”
    - Password = “password1”
  + 2. Click the confirm button and user will now see their username on the nav bar near a logout button
* User story 3 from sprint 2: As a user who isn’t signed in or doesn’t have an account, I want to be able to search for recipes and view them.
  + 1. Visit the homepage of Meal-Prep website; click on the search bar and type
    - Search = “chicken”
  + 2. Scroll down to see the results of the search which are recipes with chicken in their name. The results will have the recipe image and name.
  + 3. Click on the recipe and a modal will appear with recipe title and picture in addition to three tabs: summary, ingredients and nutrition.
  + 4 .Click on summary tab to view basic information on the recipe
  + 5. Click on the ingredients tab to see all the ingredients and their measurements in the recipe
  + 6. Click on the nutrition tab to view all the nutritional information about the recipe.
* User story 1 from sprint 3: As a user with an account, I would like to be able to view all my favorite recipes I have saved.
  + 1. Visit the homepage of Meal-Prep website; click the login button in the navbar and sign in.
  + 2.Click on the Favorites button next to the search button.
  + 3. Scroll down to see all of the recipes that have been added to favorites.
  + 4. Click on the recipes to get a modal view of the recipes with more details
* User story 2 from sprint 3: As a user with an account, I would like to be able to save recipes to my favorite recipes.
  + 1. Visit the homepage of Meal-Prep website; click the login button in the navbar and sign in.
  + 2. Click on the search bar and type
    - Search = “Beef”
  + 3. Scroll down to see all of the result recipes and click on one of them.
  + 4. A modal view of the recipe will appear with a button on it titled Add to Favorites
  + 5. Click the Add to Favorites button and you will get a notification message stating the recipe has been successfully added to your favorites.
* User story 1 from sprint 4: As a user with an account, I would like to view my grocery lists that contain recipes and ingredients from those recipes.
  + 1. Visit the homepage of Meal-Prep website; click the login button in the navbar and sign in.
  + 2. Click on the My Grocery List button on the nav bar and you will be direct the grocery list page
  + 3. A list of all the grocery list will be displayed to the user
  + 4. Click on one of the grocery lists and there will be two tabs displayed. One with the list of recipes displayed from that grocery list and a list of all ingredients from the recipes.
* User story 2 from sprint 4: As a user with an account, I would like to create a new grocery list that is added to my existing grocery list
  + 1. Visit the homepage of Meal-Prep website; click the login button in the navbar and sign in.
  + 2. Click on the My Grocery List button on the nav bar and you will be direct the grocery list page
  + Click on the Add Grocery List button and then type
    - “Grocery\_list2”
    - A notification message will be displayed stating the grocery list has been created
* User story 3 from sprint 4: As a user with an account, I would like to add new recipes to my grocery list.
  + 1. Visit the homepage of Meal-Prep website; click the login button in the navbar and sign in.
  + 2. Click on the search bar and type
    - Search = “Ice Cream”
  + 3. Scroll down to see all of the result recipes and click on one of them.
  + 4. In the modal view of the recipe, click on the button Add to Grocery List
    - Select “Grocery\_list2”
    - A notification message will be displayed stating the recipe has been added the grocery list

**Unit Tests**

**Tyler Hayden**

The module I created testing for was the RecipeViewer. This module was on the front-end of the project that would show the user all the information about the recipes. It would need to send an api call to the 3rd party api, Spoonacular, to get all the information about a specific recipe. All testing was black box testing to check that the information would be displayed with a successful and unsuccesful mocked api call. Also, I tested the functionality of the buttons in that view such as the close button and the add to favorites button. I tested the close button by passing in a mock function for the onClick event button to make sure the button would fire when it is clicked. For the add to favorites, I created a mock api call to our backend and I tested that we would get the correct notification message for a successful request and an error notification for an unsuccessful request. This allowed me to keep the testing isolated to this component alone to make sure it resulted in the correct behavior.

**Joshua Harding**

The module I mostly created testing for was the MealCalendar. It utilized the Big-Calendar React component to display clickable meal events on user specified dates. My black-box testing consisted of passing in mock recipe object calls to the meal event buttons to make sure it could correctly display meals in their specified time slots and tested Tyler’s recipe popup upon event click. I tested the MealCalendar “add” button through print statements to ensure it passed the date and times correctly. The testing for the homepage consisted of altering background edge values for both mobile and desktop, and various testing of the navbar buttons with mock routes to ensure they took users to the specified pages.

**Adam Barsness**

The modules I mainly tested were the Recipe Featured and search Results view and the Grocery List Ingredients API fetch/display (other parts of the Grocery List feature were implemented and tested by other team members). Testing methods primarily used Black Box testing, with well-placed console log statements inserted to verify the internal integrity of function and API return values. This method helped to ensure that the information passed around internally in the web application was accurate and that internal logic errors or faulty API and backend status codes were not encountered under operable circumstances.

[Shane Smith](mailto:ssmith30@ucsc.edu)

I primarily focused on CSS and the Search bar frontend functionality, CSS mostly falls into personal preference so the testing for it was minimal. Mostly I edited our applications pages to follow a design that aligned with a design the group had agreed upon to the best of my abilities. I followed up by checking how well our designed work for mobile and desktop versions of our site looked, then consulted the group and attempted to implement the agreed-upon amendments. For the Search functionality I created the search bar and tested if it was passing messages to the console.log such that I could guarantee that when another group member with the knowledge to hook it up to the API was available that they only had to test that the messages were formatted properly for the API. testing for CSS elements is forever on going as we may come across new designs in the future that we as a group wish to implement to further refine our product.

**Emmanuel Pena**

The modules I mainly tested were GroceryListTab and AddToGroceryListMenu. All testing was black-box testing. AddToGroceryListMenu was in the frontend and would fetch() grocery lists from the backend associated with the user’s account and display them in the menu options of grocery lists to add a recipe to. This module also allowed a user to add a recipe to a new grocery list by clicking on Add New Grocery List instead and then entering the new grocery list name. All existing grocery lists had to appear in the menu and in the GroceryListTab. I had mock calls to the onClick events that the module required to make sure that the behavior was as expected. The GroceryListTab displayed all the grocery list names associated with a user’s account, and if a list name was clicked, it would display a dropdown list of it’s ingredients. Also, clicking on the garbage button next to the list names would delete the lists from GroceryListTab and AddToGroceryListMenu. This also used mock calls to the onClick events that the module required in order to behave as expected. I also tested the Login module to ensure that mock calls to onClick events only logged in with correct login information.